Research Issues in Data Modeling for Scientific Visualization

Gregory M. Nielson, *Arizona State University;* **Pere Brunet**, *Polytechnical University of Catalonia;* **Markus Gross**, *Computer Graphics Center;* **Hans Hagen**, *University of Kaiserslautern;* **S.V. Klimenko**, *Institute for High Energy Physics*

What does the future hold for data visualization systems? A thumb through the trade journals or a visit to the exhibit areas of recent conferences make some predictions rather easy. The systems will become more interactive through faster response times made available through parallelism and other advanced hardware. They will allow collaborative work by geographically separated teams of experts, supported by standards and the effective use of networks. They will involve other senses, such as haptic and auditory, through the technology of multimedia and virtual reality interfaces.

While our insatiable desire for glitz and gadgetry may drive much of this development, effective analysis and visualization of large data sets will require much more than this. We envision a future where visualization systems have incorporated the ideas and technology from other disciplines.

For example, incorporating technology from database management and geographical information systems would let scientists issue such commands as "Show me all the hurricanes over Florida in 1991," or "Highlight all the objects related to hearing and supplied by this artery in this MRI scan." Incorporating techniques from numerical analysis and statistics could answer questions like "What is the volume of this tumor in the upper portion of this image?" Integrating CAD/CAM technology would support responses to such commands as "Produce a parametric surface representation of the isosurface of this left femur." AI and expert systems technology could help scientists answer questions like "What is the best way to look at the magnetic flux data over the surface of this object?"

While our vision is ambitious, we do not want to suggest that these futuristic systems will become monolithic beasts of Delphic omniscience. Clearly, the future belongs to systems with a protean configuration that use agents—perhaps similar to Internet gophers—to find alliances of data, computing resources, and problem-solving techniques that will meet the visualization task at hand. Some software projects are already adding these auxiliary capabilities to visualization systems, and software packages and systems from other areas are increasingly adding visualization tools.

But the most difficult part of integrating these capabilities effectively is not a software problem. The main purpose of data analysis and visualization is knowledge acquisition. Knowledge requires a language, and for science the language is mathematics. Modeling uses this language to describe, represent, and structure our scientific thoughts. Before we use DBMS technology to show a meteorologist a hurricane, we need to model this feature and subsequently detect it in the data. Before a cubature rule can be used to compute the volume of a tumor, we must derive a mathematical model of the tumor.

Modeling is key to the development of scientific data visualization. It is one of the "hard" research topics. Without organized, directed efforts, it is likely to be slighted, leaving us staring in frustration at beautiful, but meaningless, pictures of data. This article summarizes some topics of modeling as they impinge on the future development of scientific data visualization.

Volume modeling

Today, surfaces are the mainstay of modeling objects in computer graphics. A myriad of algorithms deal with surfaces, and many workstations are specially designed to process and render surfaces and their polygon approximations. But the future is in volumes. Volume graphics is a means to render a volume model. In the past several years, a tremendous amount of research and development has been directed toward algorithms and hardware systems for producing volume renderings, but there has been very little work on developing the volume models that feed this rendering pipeline.

Many current volume rendering applications are based on very regular and dense 3D image data from some scanning instrument, as in magnetic resonance imaging. This type of data and its constituent "voxels" are the direct 3D analog of the 2D images and pixels associated with raster graphics. We often view this data as samples (over a regular Cartesian grid) of a scalar-valued trivariate function. Some authors have used the term "volume modeling" to refer to the process of identifying and synthesizing objects contained within this type of 3D data set, but we use the term in a more general sense to mean the methods for representing and modeling the attributes of 3D objects and their interiors. The emphasis here is on the interior, whereas past geometric modeling methods have often assumed object interiors to be homogeneous.

Hanrahan[1] recently used the term "material modeling" in this regard. We do not view volume modeling as contained within volume graphics, but rather as parallel and symbiotic, similar to the relation between surface modeling and surface graphics. Just as pixel data is obtained by applying scanning algorithms to polygon approximations of surface models, we can envision algorithms that decompose a volume model into simpler constructs (the analogs of polygons) that are easily manipulated by software and hardware, further scanned to voxel data, and subsequently volume rendered (see Figure 1).

This particular pipeline view is predicated on the assumption that volume graphics will develop along the lines of discrete voxelizations and subsequent volume renderings (see Kaufman et al. in this issue, pp. 63–67), but it does not have to be this way. Just as ray tracing and radiosity algorithms can deal directly with surface models, it certainly makes sense to render a volume model directly without voxelizing it. For example, if we subscribe to a universal volume rendering integral equation that requires only a density function $\delta(x, y, z)$ and color function $C(x, y, z)$, then we can view a volume model as a representation of 3D objects and their interiors that can produce the information for defining these two attribute functions precisely.

Predicting the form of these volume models is not easy. Brunet et al.[2] briefly mentioned the very general approach of

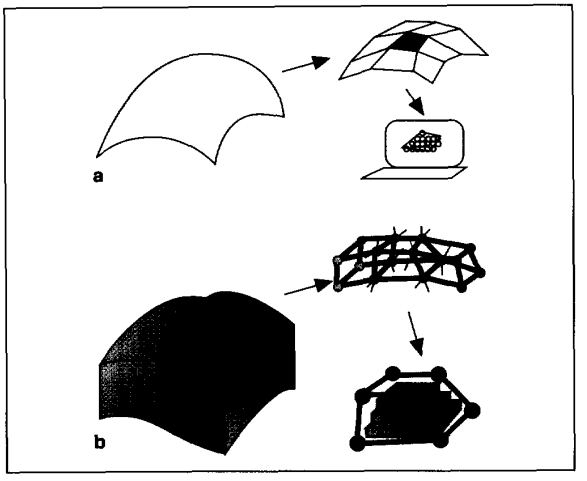

Figure 1. (a) Surface modeling and surface graphics; (b) volume modeling and volume graphics.

point topology models, which assumes attributes of objects and their interiors to be functions of their positions in space. Nielson[3] presented several other possibilities.

Multiresolution modeling

Multiresolution models can benefit scientific data analysis and visualization in many ways. For example, in browsing through a large data set, a multiresolution model can make the zoom process more efficient, thereby enhancing the chances of interactivity. This is true whether we look at isosurfaces, direct volume renderings, or topological graphs for flow visualization.

Multiresolution models can assist in geometric processing algorithms. For example, collision detection and volume intersection computations are often iterative and require reasonably good starting approximations. Some types of multiresolution models can provide these approximations.

Analysis often addresses only the general, overall structure or performance of an object. The details are not important and, in fact, may get in the way. Multiresolution models let us temporarily filter out detailed information for visualization or other analysis purposes. Also, some models contain specific qualitative information within their parameters and/or coefficients. This is the case, for example, in spectral analysis and filtering as they relate to the Fourier expansion of a signal.

Then there is compression. It seems we will never have enough bandwidth for the data we wish to work with. There is very little research in 3D data compression (see Brunet et al.[2]). Some types of multiresolution models may lead to very efficient data compression algorithms (see Gross[4]). Wavelets come to mind. Univariate wavelets form an orthogonal basis for $L^2(R)$ that are derived from a single prototype function by dilations (multiplicative scale factor) and translations. Because the wavelets are orthogonal, it is easy to compute the "best approximation" expansions; and because of the special way the wavelets are formed from the prototype function, a multiresolution analysis results from this expansion.

We can easily extend the ideas of univariate wavelets to multidimensional Cartesian grids by simply using tensor-product methods. Muraki[5] discussed some aspects of this generalization in the 3D case. Making these extensions useful within the context of data visualization poses some challenging problems, but these pale compared to the challenges of developing wavelet theory and/or multiresolution analysis for scattered data, and this is where the real potential lies.

Model validation and standards

A good portion of data visualization research addresses the development of more efficient algorithms for accomplishing a particular type of visualization. The scientific community needs a way to make qualitative and quantitative assessments of the merits of new techniques and algorithms, for example, a well-accepted set of test cases.

In some research areas, establishing a test-case set and benchmark statistics is relatively easy, but in data visualization, it is particularly difficult. One reason is the need to know the correct answer to the question, "What image should be computed from a data set?" Beyond that, how is the error between two images computed in a meaningful way? And beyond that, consider that a visualization tool's purpose is to produce not an image but rather a perception. How can we possibly put a metric on perceptions?

While arriving at accepted metrics for performance in general may be a formidable task, it might be possible to get the process started in some specific subareas. For example, in volume rendering, we can prove that the limiting value for all the "popular" volume rendering algorithms can be represented as a single equation based on two trivariate functions, δ and C, the density and color functions, respectively. These functions appear as integrands, and we can view all existing algorithms as quadrature rules for computing approximations of these integrals. This allows for some test cases where various models and algorithms can be compared to each other and validated by increasing the resolution and verifying that the results converge (under any metric) to the true answer. The question still remains of how to measure the difference between images, but this should not hinder the beginning of work in this area.

The use of standard terminology and generally accepted definitions also needs attention. Without it, communicating new ideas and concepts is difficult. Unfortunately, scientific data visualization is not yet precise in this regard. While one author may use "structured" data synonymously with "curvilinear" grid data, another would say that "text" is a good example of structured data and a collection of "video snippets" is an example of "unstructured" data. This confusing situation will get worse unless we are much more precise in the descriptions and terminology used for data sets. Most likely, this precision will come from using mathematics to define various types of data sets and data structures unambiguously.

Model-based rendering

Historically, the ideas that now fall under model-based rendering have been addressed under the topic of "gridding" and applied to bivariate scalar data. General-purpose software commonly took 2D arrays of values as input and produced a contour plot or wireframe drawing. The 2D array was assumed to contain the sampled values of a function over a regular Cartesian grid. As is often the case, however, the data of interest was not available

on a nice regular grid. Rather, it occurred at some locations of a curvilinear grid or, even worse, at scattered or irregular locations. Therefore, it was necessary to go through the gridding process and produce an array of values from original data.

Today we use the term modeling for the process of finding the parameters necessary to infer grid values from relations embedded in the data and from other information describing the data and its acquisition. One of the simplest models is based on functional relationships. In this case, a modeling function is first found that, in some sense, fits the data. The function is sampled on the type of grid required for the visualization tool, then these values are passed on to the rendering software. For volume rendering, the sampling domain would typically be a regular Cartesian grid, and a 3D array would be the data structure used to convey the gridded data.

Some research issues of model-based rendering address the development of new models for very large data sets and the collection of knowledge and experience about these models to support intelligent choices for particular classes of data set. Methods can vary considerably in the "level" of modeling that takes place, making comparisons difficult. To illustrate this, consider the volume rendering of data that is known at locations of a spherical curvilinear grid, as depicted in Figure 2. One approach uses a model based on first decomposing all the cells into tetrahedra and then assuming linear variation over each tetrahedron. The volume rendering is done by splatting each tetrahedron, which requires a visibility sort on the total collection of tetrahedra. A different approach uses a model based upon the MinNorm network spline.[6] This model samples the data over a regular Cartesian grid and passes the output to an off-the-shelf volume renderer that accepts this grid data.

Each of these methods has pros and cons, and it is not easy to compare them quantitatively. What is needed is enough usable information about these methods so that a decision can be made as to which one (possibly neither) is preferable for a particular application. It will probably be the case that determining the information required is as difficult as acquiring the information itself.

Scattered data modeling

We can view the topics under scattered data modeling as enabling technology for many other modeling areas in scientific data visualization. These topics establish the most basic techniques used in many other model-based operations. They include concepts and techniques from approximation theory and numerical analysis.

The term "scattered data" was first introduced to convey the idea of data that has no special configuration as opposed to data that, for example, might lie at the vertices of a regular Cartesian grid. This distinction is important in devising data modeling methods. For example, extending univariate methods to higher dimensions is usually easy if the data lies on a Cartesian grid, but may be difficult otherwise.

It is common to divide methods of scattered data modeling into two classes: distance-weighted and cellular decomposition. Typical examples of distance-weighted methods for volumetric data are volume splines, the 3D version of the Modified Quadratic Shepard (MQS) method, and multiquadrics (see Hagen[7] and Nielson[6]). The volume spline and multiquadric methods work quite well for small data sets (fewer than 500) and are easy to implement. The MQS method works for much larger data sets, but it is harder to implement and requires certain user-defined parameters for optimal performance.

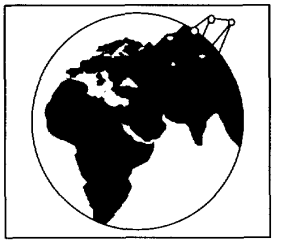

Figure 2. Spherical curvilinear grid data.

Examples of cellular decomposition methods for volumetric data are the 3D version of the Minimum Norm Network (MNN) spline[6] and the localized versions of the volume spline. The MNN can be applied to very large data sets, but its implementation requires a tetrahedrization algorithm and a fairly complicated iterative method for solving a large, sparse equation system. There are general strategies for localizing methods, but these approaches still need considerable work before they are really viable.

Research is needed to develop simple, efficient, accurate, and easily implemented modeling methods for very large data sets. Estimating and controlling errors is of particular interest. So is developing manifold methods whose domains are more general than a simple region of Euclidean space. For example, many applications require spherical domains or a surface domain (say, of a wing).

Model-based segmentation

We have borrowed the term "model-based segmentation" from Hanrahan.[1] Segmentation is one step along the path to extracting meaningful objects from acquired physical data. The term segmentation usually applies to medical data where it means the process of identifying which pixels or voxels of an image belong to a particular object such as air, bone, fat, or tissue. Of course, this same segmentation and feature-extraction process is of potential interest for other kinds of data, such as identifying the geometry of certain subterranean objects in geophysical data.

There are many other application areas. In fact, the information or knowledge to be gained from measured data often relates to the specification and description of objects contained in the data. Model-based segmentation uses mathematical models to detect and represent these objects. The ideas of constraint topology or feature modeling are similar and useful in this context. Here, certain properties and features are built into the underlying model. Then the model uses the particular data at hand to select the parameters through an optimization process. One way to factor in user expertise is to employ expert system technology in the selection and definition of the error norms to be optimized.

Conclusion

The benefits from visualization techniques in analyzing data are well established, but to build on these pioneering efforts, we must recognize modeling as a distinct structural component in the larger context of visualization and problem-solving systems. Volume modeling is the entryway to this arena of future development, and model-based rendering describes how scientists will view the results. Important side developments such as multiresolution modeling and model-based segmentation will contribute structural capability to these systems. All of these components ultimately depend on the mathematical foundations of scattered data modeling and on model validation and

standards to incorporate this modeling methodology into effective tools for scientific inquiry. ❑

References

1. P. Hanrahan, "Future Directions in 3D Medical Imaging," Course 21, Siggraph 93, ACM Press, New York, 1993, pp. 135-141.
2. P. Brunet et al., "Modeling and Visualization Through Data Compression," to appear in *Proc. ONR Workshop on Data Visualization*, Academic Press, New York, 1994.
3. G.M. Nielson, "Research Issues in Modeling for the Analysis and Visualization of Large Data Sets," to appear in *Frontiers of Scientific Visualization*, Academic Press, New York, 1994.
4. M.H. Gross, "Subspace Methods for Visualization of Multidimensional Data Sets," to appear in *Proc. ONR Workshop on Data Visualization*, Academic Press, New York, 1994.
5. S. Muraki, "Volume Data and Wavelet Transforms," *IEEE CG&A*, Vol. 13, No. 4, July 1993, pp. 50-56.
6. G.M. Nielson, "Modeling and Visualizing Volumetric and Surface-on-Surface Data," in *Focus on Scientific Visualization*, H. Hagen, H. Mueller, and G. Nielson, eds., Springer, Berlin, 1993, pp. 219-274.
7. H. Hagen, "Visualization of Large Data Sets," to appear in *Proc. ONR Workshop on Data Visualization*, Academic Press, New York, 1994.

Research Issues in the Foundations of Visualization

Philip K. Robertson, *CSIRO;* Rae A. Earnshaw, *University of Leeds;* Daniel Thalmann, *Swiss Federal Institute of Technology;* Michel Grave, *ONERA-DMI;* Julian Gallop, *Rutherford Appleton Laboratory;* Eric M. De Jong, *Jet Propulsion Laboratory*

Many visualization tools and techniques used by scientists are integrated to some degree within a system. Few systems, however, fully meet their users' needs. Limited functionality, limited information about implicit assumptions or embedded constraints, and incomplete integration of different tools are all common problems. These limitations arise partly from the historical evolution of proprietary or application-specific systems but also from the lack of clear articulation of scientific visualization's foundation assumptions and practices. Applying these assumptions and practices systematically when building tools or validating visualizations also requires articulating them clearly. We focus here on the research required to establish a foundation for the evolving needs of visualization systems, as determined during the Office of Naval Research workshop discussions. We believe that three main topics underpin these needs:

- Models: the need for abstractions to describe the core components of the visualization process and the interfaces between them, including users and their behavior.
- Validation: the problem of determining whether visualizations meet consistency and effectiveness criteria on test data or measures.
- Systems: the design, realization, and operational problems of systems integrating a range of functionalities to give scientists a working environment for visualization.

We outline key aspects of each topic below, commenting on the current status of work and isolating areas that require significant research. We conclude by suggesting strategies to initiate this research.

Models

Further progress in visualization systems will be difficult without more formally addressing the "models," implicit or explicit, that underly current systems.[1] Therefore, we give highest priority to the need for models. We also believe that flexible models can clarify validation and system design requirements within a unified framework.

There is a clear need for an overall reference model of the visualization process, but we also need models of key components for independent use. For example, we consider a data model absolutely essential to writing new application software that can apply across different environments. We also believe a formal model of time is required to support accurate tracking and alignment of different time-dependent information. Third, a user model can clarify expectations, assumptions, and implications relating to specific and general users' interactions with visualization systems.

Reference model

A formally defined reference model can separate components of the visualization process by identifying core functionalities. It can also serve as a basis for standardizing terminology, comparing and choosing systems, and identifying constraints or limitations in our current understanding of the process. There is currently no widely accepted reference model that meets these requirements. As a consequence, terminology varies and comparisons are difficult. Some partial models have been proposed. For example, the dataflow pipeline is a widely used model, but it does not fully describe even the systems currently in use and their integration with simulation and modeling. Further, it lacks a recognized formal description. There is thus a strong need for an initial reference model with terminology definitions and sufficient flexibility to evolve as the field matures.[2]

Data models

Despite widely available tools to convert between the many data formats used in visualization, and despite many calls for data models to support application development,[3,4] the community has not yet established models that adequately describe the full range of data used. It is not enough simply to specify data dimensionality.

Data models have been proposed for specific data types and characteristics.[5] If we expect applications to handle data at increasingly semantic levels, we need models that can fully describe data at a generic level. Such a model could also clarify—for users and applications—which processes can be applied to what data.

The ability to interact at a level of abstraction from the generic structure of the data, while maintaining full integrity of the data structures, can free applications developers from